

IEEE-1588™ STANDARD FOR A PRECISION CLOCK SYNCHRONIZATION PROTOCOL FOR NETWORKED MEASUREMENT AND CONTROL SYSTEMS

John C. Eidson

Agilent Laboratories, 3500 Deer Creek Rd., Palo Alto, CA, 94304
650-485-4263 (phone), john_eidson@agilent.com

Michael C. Fischer

Fischer Consulting, 763 E. Charleston Rd., Palo Alto, CA, 94303
650-855-9816 (phone), mikecf@jps.net

Joe White

U.S. Naval Research Laboratory, Washington, D.C.
202-767-5111 (phone), joe.white@nrl.navy.mil

Abstract

This paper will discuss the major features and design objectives of the IEEE-1588 standard. Recent performance results of prototype implementations of this standard in an Ethernet environment will be presented. Potential areas of application of this standard will be outlined.

INTRODUCTION

Temporal relationships have always been an important element in the measurement and control of industrial physical systems. In small closed systems time is usually implicit in the operation of electronic circuits or in the execution patterns of computer programs. As these *industrial* systems become more complex with sensors, actuators, and computers distributed in space and communicating via networks, the explicit representation of time is often necessary for robust implementations. The temporal and other implementation requirements on industrial systems differ considerably from those found in typical office distributed computing environments. IEEE-1588-2002, ‘Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems’ was designed to serve the clock synchronization needs of industrial systems.

TEMPORAL REQUIREMENTS FOR INDUSTRIAL APPLICATIONS

Application targets for IEEE-1588 are systems typically found in laboratory or in product test, industrial automation, motion control, power or telecommunications system installations and similar industrial settings involving multiple sensors, actuators, instruments, and computer/controllers. Temporal, or synchronization, requirements in these applications are typically met in one of three ways:

1. Message-based. In message-based timing the sensing of a datum, the setting of an actuator, or the initiation of a control procedure is synchronized based on the event of receiving a command or message. IEEE-488 instrument systems and many industrial control systems based on proprietary

networks are good examples. In these systems precise control of communication latency and execution timing is required for accurate system wide synchronization.

2. **Cyclic-systems.** In cyclic-systems timing is periodic and is usually defined by the characteristics of a cyclic network or bus, for example the SERCOS bus widely used in the motion control industry. Synchronization accuracy depends of the accuracy of the cycle period and on the latency and fluctuations introduced by participating devices in converting the cycle timing into the required actions.
3. **Time-based.** In time-based systems the execution of events is based on a common sense of time. For each event an execution time is specified with the event execution occurring when the specified time matches the measure of real-time. This is the most flexible of the three schemes in that different, and if needed incommensurate, timing schedules for each device are easily implemented. In addition synchronization accuracy depends on the accuracy of the common sense of time and on the implementation of the participating devices rather than on precise control of messaging latency.

For the most accurate synchronization in a time-based system, the common sense of time will be implemented by having a local clock in each participating node synchronized to its peers via a protocol such as IEEE-1588. The required accuracy in synchronizing these clocks depends on the application. Typical applications and their required accuracies are listed in Table 1.

Table 1: Typical application synchronization requirements

Application area	Required synchronization accuracy
Low speed sensors (e.g. pressure, temperature)	Milliseconds
Common electro-mechanical devices (e.g. relays, breakers, solenoids, valves)	Milliseconds
General automation (e.g. materials handling, chemical processing)	Milliseconds
Precise motion control (e.g. high speed packaging, printing, robotics)	A few microseconds
High speed electrical devices (e.g. synchrophasor measurements)	Microseconds
Electronic ranging (e.g. fault detection, triangulation)	Sub microsecond

In addition to synchronization requirements the targeted application areas typically include one or more of the following additional requirements:

1. **Networking.** Distributed systems increasingly use networks such as Ethernet for communication. However there is still a need to provide a common sense of time on other networks.
2. **Heterogeneous systems.** Most systems have a range of synchronization accuracies. Therefore the synchronization protocol must accommodate devices with varying accuracy capabilities.
3. **Cost.** Systems will include both high and low cost devices. Low cost devices typically will have limited computational capability and memory available to execute a synchronization protocol.
4. **Spatial scale.** Most systems are compact enough both physically and logically to be implemented in a few local subnets of the communication system. Larger scale application can usually be treated as islands of locally precise time in which a separate protocol, for example GPS, is used for inter-island synchronization.
5. **Low administrative overhead.** In the simplest case, the protocol should be self-configuring.

IEEE-1588 is designed to satisfy all of these requirements. It is not designed to work over the Internet or the general computing environment typically served by the popular Network Time Protocol [4].

REQUIREMENTS FOR MILITARY APPLICATIONS

While military applications were not explicitly considered during the development of IEEE-1588 it appears that many of the characteristics discussed above for industrial systems are applicable to the newer generations of military systems. In particular military systems are evolving from stand-alone systems into architecture of interoperable systems with strong synchronization or coordination requirements. A common sense of time is to be the foundation for this architecture with the worldwide GPS system as the

key component. However operation must be insensitive to the loss of GPS signals, [6]. At least in local environments such as a ship, or command center a system based on IEEE-1588 may allow more robust local operation by maintaining local time consistency even when isolated from the global system. As system requirements extend to smaller, less capable and cheaper military devices the low implementation footprints for IEEE-1588 may be advantageous as well.

Further discussion of applications of IEEE 1588 may be found other papers, [1], [5]. The remainder of this paper discusses performance experience and how IEEE-1588 is structured to meet timing requirements.

IEEE-1588 DESIGN FOR MEETING TEMPORAL REQUIREMENTS

Within a subnet IEEE-1588 automatically establishes a master-slave relationship among the participating clocks communicating via the subnet. The master is selected as the best clock based on defined descriptors for inherent accuracy, traceability to UTC, etc. Provision is made to designate a set of clocks to be preferred in this selection for applications where this is important.

The slaves synchronize their local clocks to that of their master by an exchange of messages illustrated in Figure 1. Periodically the master clock sends a distinguished message, a Sync message, as a multicast to all its slaves. This message contains an estimate of when the Sync message will be placed on the network. In the most accurate implementations the master will contain a mechanism for detecting and time stamping based on the master's local clock, the time that the Sync message is actually placed on the network. In Ethernet an ideal place to attach this detector is at the MII interface of the PHY chip, thus avoiding the large temporal fluctuations in the upper portions of the protocol stack. If the master is so equipped, the master's IEEE-1588 code sends this measured time stamp to all slaves in a second message, the Follow_up message. The slaves receive the Sync message and if so equipped detect and time stamp its arrival as close to the network as possible. Upon receiving the Sync message and for highest accuracy the Follow_up message and the local receipt time for the Sync message, the slave's IEEE-1588 code uses this information to correct the time of the slave's local clock. Periodically, but with longer period to reduce network loading this process is reversed. This forward and reverse path information is used to compute the one-way network latency on the assumption that the path is symmetrical. The slaves use this measured latency in computing the correction to their local clock. This procedure effectively removes timing fluctuations in the participating end devices and latency in the communication path.

The other major source of timing errors is network latency fluctuations introduced by network elements. In an Ethernet environment this includes repeaters, switches, and routers. Of these, routers introduce fluctuations too large and inconsistent to be reduced to the desired accuracy with statistics. IEEE-1588 specifies a transfer standard mechanism, the boundary clock, for logically eliminating routers from the IEEE-1588 protocol communication path. The boundary clock appears to each subnet of interest as an ordinary IEEE-1588 clock as provided in an end device. However the boundary clock uses a single local clock to serve as a master in either all or all but one of the subnets of interest. In the rare case where multiple routers are needed in an application, the boundary clocks themselves form a hierarchy so that in a properly implemented IEEE-1588 system there is always a single clock serving as the primary source of time for the ensemble. The time base of this clock may be synchronized to a reliable source of UTC time if required.

In the restricted environments typical of the target applications, the fluctuations in repeaters and usually in switches can generally be reduced to acceptable levels by the use of statistical techniques amenable to low cost devices. Modern switches implement level 2 protocols that can further reduce fluctuations based on the priority of Sync messages [3]. It is also possible to design switches incorporating IEEE-1588

boundary clocks that logically remove them from the IEEE-1588 communication paths. This technique effectively produces direct synchronization between the clocks in the end device and the switch.

Further details on these mechanisms and other features of IEEE-1588 beyond the scope of this paper may be found in the standard itself or on the IEEE-1588 web site [5].

IEEE-1588 PERFORMANCE EXPERIENCE

This section discusses performance measurements on a prototype implementation of IEEE-1588 under varying network topologies. The measurements are made as illustrated in Figure 2. The ASIC shown contains a 68020 class 40Mhz processor, a Sync message detector observing the MII interface as illustrated in Figure 1, and a hardware real-time clock with a resolution of 25 ns. Each clock has a 1 PPS test point that rolls over at the seconds boundary of the clock. An Agilent 5372A Frequency and Time Analyzer is used to measure the relative offsets between the 1 PPS signals of the master and slave. In each case the data represents statistics on 3600 measurements made over a one-hour period.

The prototypes use a simple PI control loop in the slave clocks to servo the local clock to that of the master. The clock offset errors are sampled every 2 seconds as the basis for the PI loop input. Two sets of parameters for the proportional and integral terms in the PI loop are reported. A *fast loop* set with $P=2$ and $I = 0.5$, and a *slow loop* set with $P=0.5$ and $I=0.125$. The fast loop has relatively high gain, wide bandwidth and fast response time, a desirable condition during startup to speed acquisition of lock and to drive large initial errors to small values, typically on the order of a minute.

In addition the slaves implement a statistical algorithm to discard from consideration by the control loop obvious outliers in the measured delay and offset between the master and slave. The data reported here was collected with this algorithm disabled to allow observation of the underlying performance of the system.

The hardware clocks are driven by 40 MHz crystal oscillators. Data derived using two different grades of oscillators is reported. The Allan frequency deviations of these oscillators are shown in Figure 3. The CTS CB3LV, *inexpensive oscillator*, is typical of oscillators used in cost sensitive applications. The 10811D oscillators are ovenized, instrument grade oscillators. Based on the data in Figure 3 the expected time fluctuations introduced by the oscillators are shown in Table 2 for several values of the averaging time τ .

Table 2: Expected oscillator induced time fluctuation

τ - seconds	Fluctuations- nanoseconds	
	Inexpensive	10811D
1	4	0.006
2	9.2	0.006
10	90	0.040
100	1400	0.700

DIRECT CONNECTION BETWEEN CLOCKS

The direct connection of two clocks, e.g. connected via a crossover cable rather than via a repeater or switch, was tested to show the limiting characteristics of the clock implementation free from fluctuations introduced by network components. This case also represents the expected performance for topologies in

which end devices interact directly with a switch, or any other device, implementing IEEE-1588 boundary clock functionality.

Figure 4 contains histograms of the synchronization error for two clocks directly connected via Ethernet for both inexpensive and 10811D oscillators and for the slow and fast loop parameters. Figure 5 illustrates approximately two-minute segments of the time variation of the clock offsets giving rise to these histograms.

The time offset waveforms with the 10811D oscillators are consistent with the hunting or limit cycle, behavior introduced by the 25 ns quantization of the two clocks, the resolution and operation of the digitally controlled clock rate adjustment of the devices and the loop parameters. No fluctuations from the 10811D oscillators are significant for the range of parameters used as noted in Table 2. With the inexpensive oscillators and the fast loop (a few seconds time constant) the clocks manage to track the oscillator induced fluctuations well enough to keep the offsets reasonably well bounded as illustrated by the width of the histogram. However with the slow loop the fluctuations of the inexpensive oscillators become more apparent as seen in the increased width of the histogram. The means and standard deviations of the measurements in Figure 4 are shown in Table 3. With the 10811D oscillators one would expect the standard deviations to improve in implementations with finer clock resolution and with a true VCO rather than the digital version of the reported implementation.

Table 3: Offset error statistics for direct connection

Loop speed	Oscillators	Mean-ns	Std.Dev-ns
Fast loop	Inexpensive oscillators	-28	34
	10811D oscillators	-1	39
Slow loop	Inexpensive oscillators	-21	76
	10811D oscillators	5	15

CLOCKS CONNECTED VIA A REPEATER

For these measurements the clocks were connected via a single HP J4090A Ethernet repeater. No other traffic other than synchronization messages was present on the subnet.

Figure 6 contains histograms of the synchronization error for the repeater connected clocks for both inexpensive and 10811D oscillators and for the slow and fast loop parameters. The means and standard deviations of the measurements in Figure 6 are shown in Table 4.

Table 4: Offset error statistics for connection via a repeater

Loop speed	Oscillators	Mean-ns	Std.Dev-ns
Fast loop	Inexpensive oscillators	-27	75
	10811D oscillators	-7	46
Slow loop	Inexpensive oscillators	-32	80
	10811D oscillators	10	42

In this case the repeater introduced measurable fluctuations for all cases resulting in the increase in the widths of the histograms compared to the corresponding cases for the direct connection.

CLOCKS CONNECTED VIA A SWITCH

For these measurements the clocks were connected via a single HP J4121A Ethernet switch. No other

traffic other than synchronization messages was present on the subnet.

Figure 7 contains histograms of the synchronization error for the switch connected clocks for both inexpensive and 10811D oscillators and for the slow and fast loop parameters. The means and standard deviations of the measurements in Figure 7 are shown in Table 5.

Table 5: Offset error statistics for connection via a switch

Loop speed	Oscillators	Mean-ns	Std.Dev-ns
Fast loop	Inexpensive oscillators	-49	140
	10811D oscillators	-14	138
Slow loop	Inexpensive oscillators	-21	123
	10811D oscillators	5	92

Comparing the data for the repeater and the switch, the switch clearly introduces more fluctuations.

SUMMARY AND ANALYSIS OF MEASUREMENT RESULTS

The standard deviations for the various cases are summarized in Table 6.

Table 6: Offset standard deviations-ns summary

Loop speed	Connection	Inexpensive oscillators	10811D
Fast loop	Direct	34	39
	Repeater	75	46
	Switch	140	138
Slow loop	Direct	76	15
	Repeater	80	42
	Switch	123	92

The distributions for the various cases are not sufficiently close to normal to justify detailed calculations. However qualitatively it is clear that the fluctuations due to the repeater and switch either dominate or at worst are the same order of magnitude as the inherent characteristics of the nodes themselves. Further the results for both repeater and switch are relatively independent of the loop parameters used in the experiments although the switch perhaps exhibits smaller effects with the slow loop. It is clear that the differences in the direct connection case indicate that the choice of loop parameters should be optimized for the particular oscillator in use. Although not reported here, prior work has shown fluctuations roughly a factor of 2 to 3 higher in the case of repeaters and switches carrying moderate traffic on our laboratory LAN, [5].

SUMMARY

We reported results on prototype IEEE-1588 clocks under a variety of network topologies and implementation parameters. This data indicates that IEEE-1588 will be able to support sub-microsecond synchronization in a variety of form factors suitable for the target markets. The standard has been approved by the IEEE and may be ordered via the IEEE Standards web site [2].

REFERENCES

[1]: ‘Synchronizing Measurement and Control Systems’, John C. Eidson, R. William Kneifel II, Anatoly Moldovansky, and Stan P. Woods. Sensors Magazine, November 2002.

[2]; See <http://standards.ieee.org/>

[3]: 'Utilization of Modern Switching Technology in EtherNet/IP™ Networks', Anatoly Moldovansky, RTLIA 2002, 1st Int'l Workshop on Real Time LANS in the Internet Age, Technical University of Vienna, Austria, June 18, 2002

[4]: See <http://www.eecis.udel.edu/~ntp/> for details concerning the Network Time Protocol.

[5]: See 'IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems' at <http://ieee1588.nist.gov>

[6]: "Common Time Reference Technology for Systems Interoperability", R. Beard and J. White, U.S. Naval Research Laboratory, Proceedings of the 32nd Precise Time and Time Interval Systems and Applications Meeting, 2000, pp 415-422.

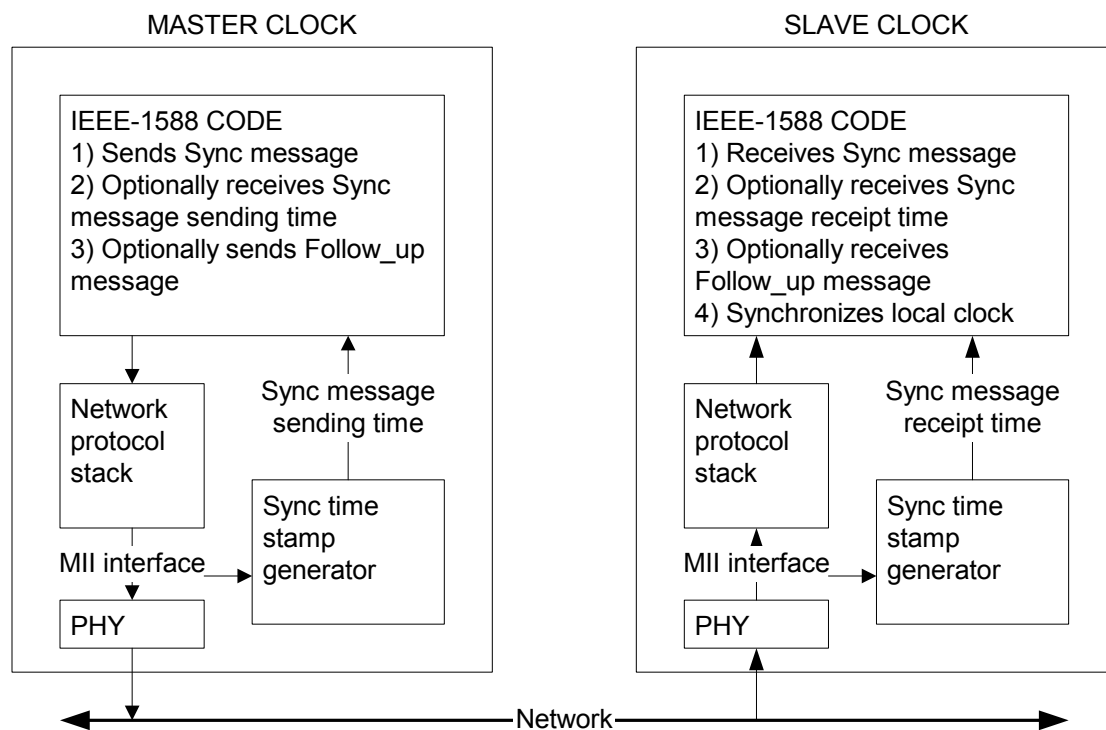


Figure 1: IEEE-1588 messages used in synchronization

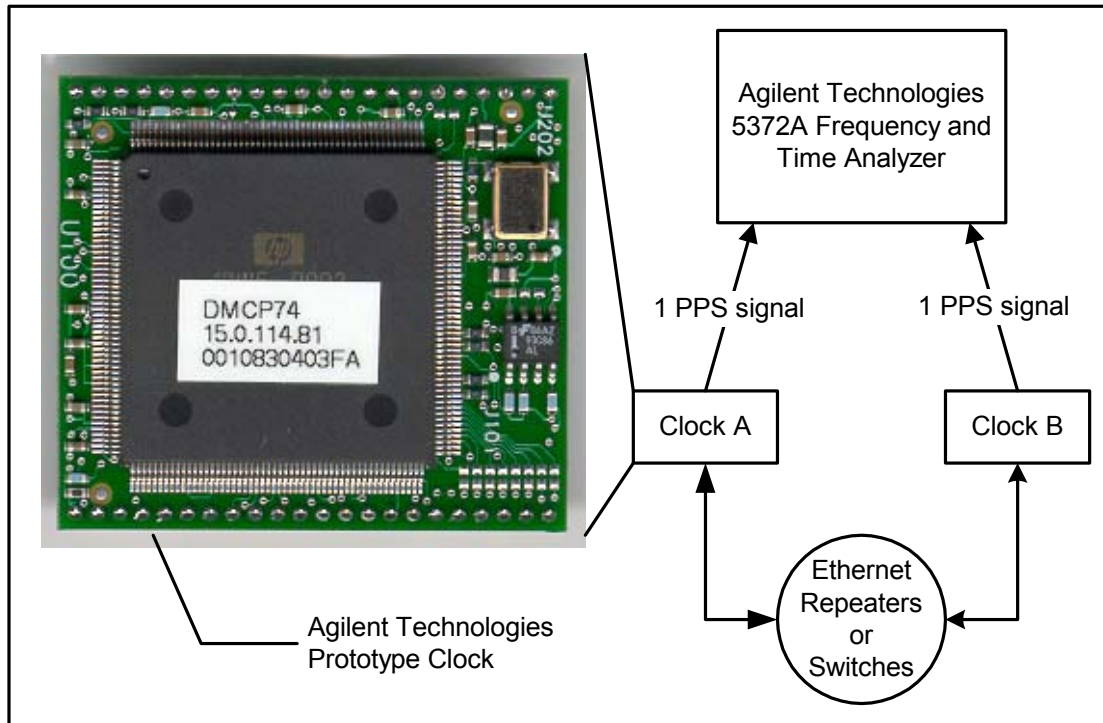


Figure 2: Measurement topology

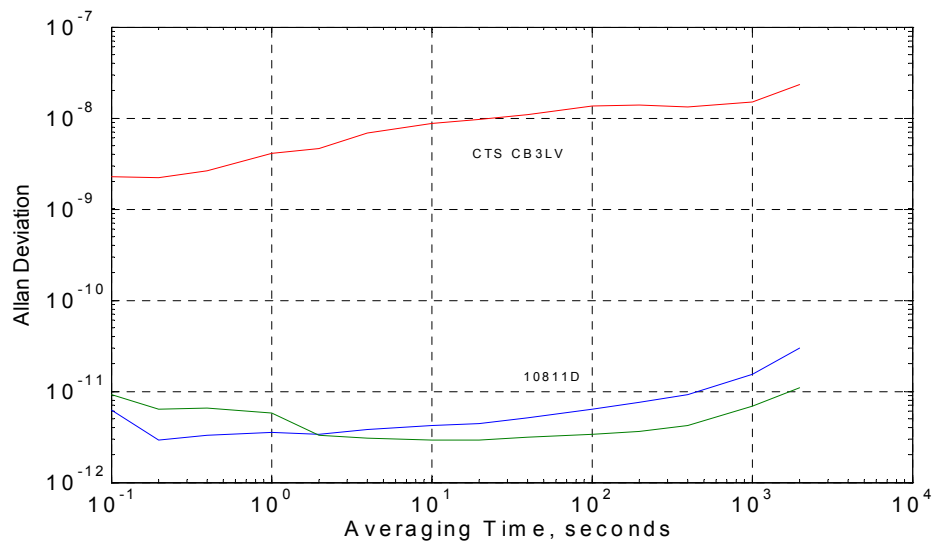


Figure 3: Allan frequency deviations for test oscillators

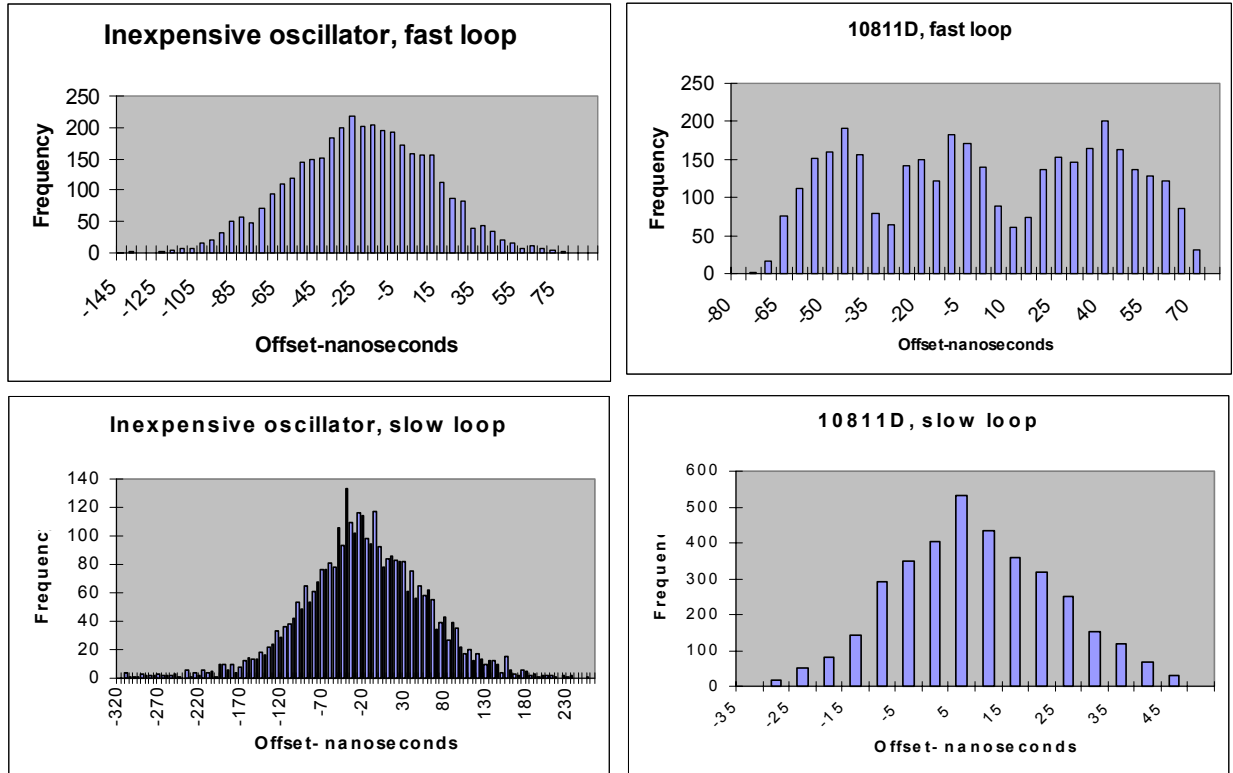


Figure 4: Histogram of synchronization error for direct connection

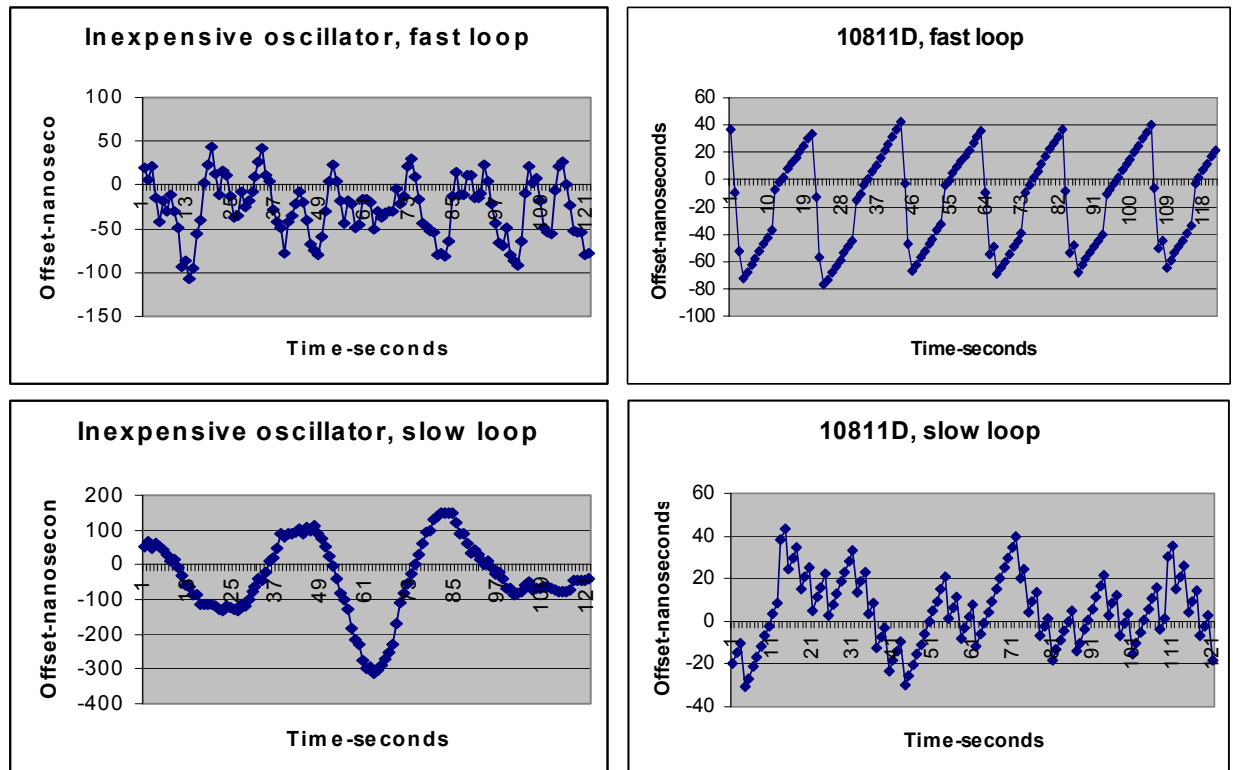


Figure 5: Synchronization error for direct connection

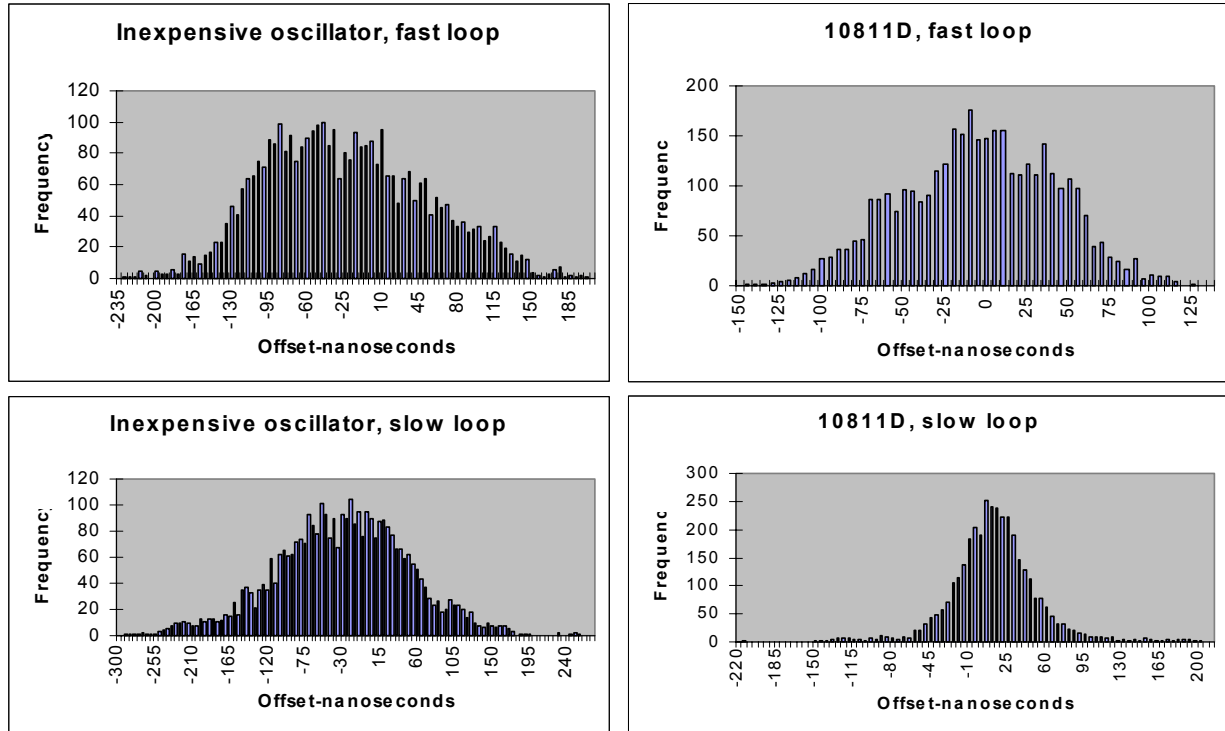


Figure 6: Synchronization error for connection via a repeater

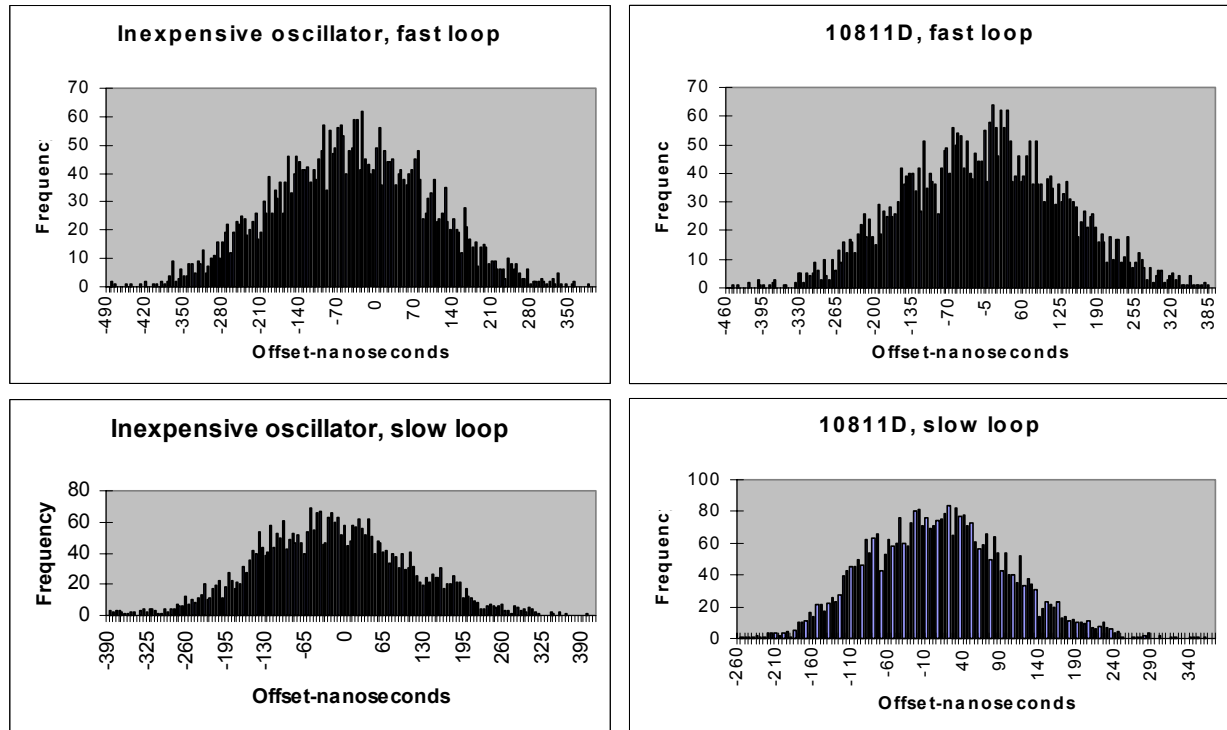


Figure 7: Synchronization error for connection via a switch